WASAM: A Dynamic Wide-Area Situational Awareness Model for Critical Domains in Smart Grids

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

Control from anywhere and at anytime is nowadays a matter of paramount importance in critical systems. This is the case of the Smart Grid and its domains which should be monitored through intelligent and dynamic mechanisms able to anticipate, detect and respond before disruptions arise within the system. Given this fact and its importance for social welfare and the economy, a model for wide-area situational awareness is proposed in this paper. The model is based on a set of current technologies such as the wireless sensor networks, the ISA100.11a standard and cloud-computing together with a set of high-level functional services. These services include global and local support for prevention through a simple forecast scheme, detection of anomalies in the observation tasks, response to incidents, tests of accuracy and maintenance, as well as recovery of states and control in crisis situations.

Keywords: Critical Infrastructure Protection, Wide-Area Situational Awareness, Smart Grid, Wireless Sensor Networks, Cloud Computing

1 Introduction

A Smart Grid is a complex distributed infrastructure composed of a set of domains and stakeholders. According to the conceptual model of the National Institute of Standards and Technology (NIST), these domains correspond to customers, markets, providers, energy generation, distribution and transmission networks, as well as control systems such as SCADA (Supervisory Control and Data Acquisition) systems [1, 2]. This last domain can be considered as the core of the entire system that widely interconnects with the other domains. This interconnection enables the SCADA Center to know the performance of the entire Grid and control its functions for delivering essential services, such as electrical energy. However, this control mostly depends on a set of technologies and communication systems, highlighting amongst them, Wireless Sensor Networks (WSNs) for smart supervision or cloud computing as an alternative support for redundancy and availability of resources, services and data at a low cost in management. This alternative support also guarantees high scalability, resilience and recovery capabilities when parts of the system are isolated, inaccessible or lost [3]. For example, if the control of a SCADA system is (temporarily or permanently) lost, another SCADA system may retake control through the ICCP (Inter-Control Center Communications Protocol) industrial protocol using information stored within the cloud [3]. This can happen at control substations in charge of supervising in real-time, the performance and functionality of energy generation, transmission and distribution systems. These substations have a tendency to experience numerous and unforeseen events caused by failures/errors, which may even provoke instabilities that could trigger a devastating cascading effect with a high probability of reaching other domains within the Grid.

We agree with NIST that the topic of Situational Awareness (SA) for the control of large geographic distributions should be a priority topic within protection. In fact, NIST classifies it as one of the eight priority areas to be researched, calling it: *Wide-Area Situational Awareness* (WASA) [1]. To address this new paradigm of protection, the work [4] exhibits an abstract methodological framework for SA based on a set of preventive and reactive procedures for the protection, and addresses several conceptual standpoints for WASA. Some researchers believe that SA for dynamic and complex systems should be related to the cognitive model [5] based on the perception of physical elements (e.g., levels of voltage) in an environment, the interpretation of their meaning, and the projection of their status in the near future. In contrast, other researchers believe there is a need to build hybrid models where human knowledge is fundamental for obtaining a more objective vision within the context [4]. This is the case of the model proposed in [6] where industrial communication traffic is analysed in order to verify and report on the integrity of the observed system at a high level of comprehension for human operators. However, and unfortunately, more investigation in this field is still required so as to offer an improved overview of the situation, where the system can also anticipate and respond to threatening situations. Given this, we therefore propose, in this paper, a dynamic model, called WASAM *(Wide-Area Situational Awareness Model)*, based on the use of different technologies to ensure control at all times, in addition to offering a support for WASA that complies with the methodological framework given in [4].

The proposed model is based on: (i) the technology of WSN for monitoring; (ii) the ISA100.11a standard [7] for managing different kinds of SCADA incidents; (iii) two preventive methods, one of them focusing on anticipating infrastructural anomalies (control of physical events of the observed infrastructure) and the other on anomaly control (malfunctions) within the control network; (iv) a self-validation mechanism to evaluate the real state of the proposed model; and (v) the cloud technology based on Sensitive Data (SD) for future governance aspects and recovery purposes. This last integration, also known as cloud-based SCADA, allows different system elements to on-demand gain accesses to a shared pool of the resources supplied, such as networks, servers, applications, storage and services. For example, Mohsenian-Rad et. al consider this infrastructure as a suitable massive data center for critical systems where backup instances can strategically be disseminated within the cloud [8]. This feature is also considered by the CloudCERT project [9], which aims to extend the advantages of the technology by offering the necessary tools for information exchange in emergency situations. Nonetheless, more investigation is still needed in the critical infrastructure protection field, and more particularly, in the provision of predictive and reactive solutions.

The paper is organized as follows. Section 2 introduces the basic components for the construction of the model, which will be used later for the design in Section 3. The model and its components, technologies and methods for prevention, detection, response and self-validation are discussed in detail in Section 3.1 and Section 3.2. Finally, Section 4 states some results of the simulation, and Section 5 concludes the paper and outlines future work.

2 Construction Components for WASAM

A WASA system should comprise advanced monitoring components with integrated techniques that help analyse and interpret data streams, in addition to guaranteeing aspects of extensibility for its application in a future. Given this, four main components should be considered for the construction of the model: (i) a *detection component*, (ii) a *recollection component*, (iii) an *alarm management component* to issue alerts and warn the system, and (iv) a *reaction component*. The detection component is based on sensory devices able to monitor physical events, detect and track behaviour, and warn the gateway of anomalous situations to the gateway [10]. The gateway is a powerful device that serves as an interface between the acquisition world (i.e., the WSN) and the real world (i.e., the SCADA Center). In addition, these sensor nodes are smart devices with the capability of collaborating with each other and they are able to guarantee self-configuration in order to adapt themselves to the conditions of the network. With respect to the recollection component, it is represented by the SCADA Center itself, the SD cloud and any external storage device in charge of registering SCADA evidence flows. The use of cloud computing for evidence storage enables the system to maintain backup instances at different locations within the cloud and in a balanced manner, in addition to guaranteeing redundant configurations to restore previous states and ensuring safety-critical (in operational terms) in crisis scenarios. A safety-critical is considered an essential property [11] that should be considered when the underlying infrastructure is critical and any unplanned event may potentially trigger a cascading effect.

The alarm management component is based on specific management systems offered by existing wireless industrial communication standards, such as

ISA100.11a. This standard provides a set of services for communication reliability, security (based on symmetric/asymmetric cryptography), coexistence, and priority-based alarm management using up to five criticality levels: *journal, low, medium, high* and *urgent*. Its networks can support sensor nodes working at 13-180MHz, 256-512KB RAM, 4-32MB ROM and 40mA of energy, and one or several gateways to establish redundant connections with the SCADA Center. The information from the sensors is managed through DMAP (Device Management Application Process) objects. DMAP is a class installed inside each device, which includes a set of objects used for configuring, supervising and requesting network parameters. More specifically, DMAP contemplates the ARMO (Alert Reporting Management Object) class for managing alerts and generating reports through an AlertReport service to ARO (Alert Receiving Object). ARO is a class configured in only one device in the network; the gateway in our case. Finally, the reaction component carries out decision-making processes that depend on a set of factors, such as the simplicity of the technique applied, which should not increase functional complexities, and the autonomous and dynamic capacity of the model to address threatening situations. In our case, this component is principally based on a set of integrated modules that collaborate with each other to carry out several tasks. Some of them are; to estimate the proximity of a possible anomaly, locate and warn the nearest operator in the area, evaluate the level of accuracy in the detection and prevention tasks, and frequently report the real state of the network.

3 The proposed model

As ISA100.11a permits the configuration of diverse types of networks, the architecture of the model (See Fig. 1) is based on a hierarchical configuration; where nodes are grouped into clusters and all the organizational decisions are carried out by a special entity known as the *Cluster Head* (CH). Each CH$_j$ is responsible for receiving and checking information (either readings or ISA100.11a alarms) from their sensors in order to detect and warn of anomalous conduct through patterns, in addition to filtering and aggregating information (main tasks of a CH) to be resent to the gateway later. There are three reasons for selecting this configuration. Firstly, this configuration allows the system to efficiently manage its resources since CH devices typically have greater computational capabilities than other nodes [10]. Secondly, it is possible to rapidly locate anomalies when the network deployment is known in advance. Finally, part of the processing is made straightforward by using simple behaviour patterns.

An anomalous behaviour can be defined as "*something deviated from what is standard, normal, or expected*". From this definition, taken from the Oxford Dictionary [12], we deduce that if a reading is not inside a prescribed threshold ($[V_{min}, V_{max}]$) defined by the SCADA organization, electrical companies or countries, then it can be considered anomalous. As our model measures infrastructural anomalies related to significant changes of voltage readings (denoted as v_i), a deviation from the allowable thresholds is therefore considered as an

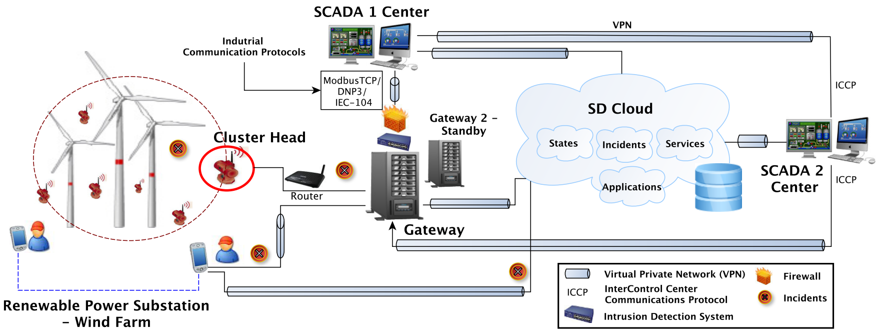

Figure 1: General Architecture of the WASAM Model

anomaly. When this situation appears, the system has to deliver an alarm. Taking advantage of ISA100.11a and its alarm management, we can consider three principal situations: *normal situation* (valid readings where $v_i \in [V_{min}, V_{max}]$ which is signaled with the state 0), *unstable situation* (non-critical alarms where $v_i \notin [V_{min}, V_{max}]$ that do not compromise the security/safety of the system, with priority values of journal (1), low (2) and medium (3)), and *critical situation* (critical alarms where $v_i \notin [V_{min}, V_{max}]$ that may compromise the security of the system with priority values of high (4) and urgent (5)).

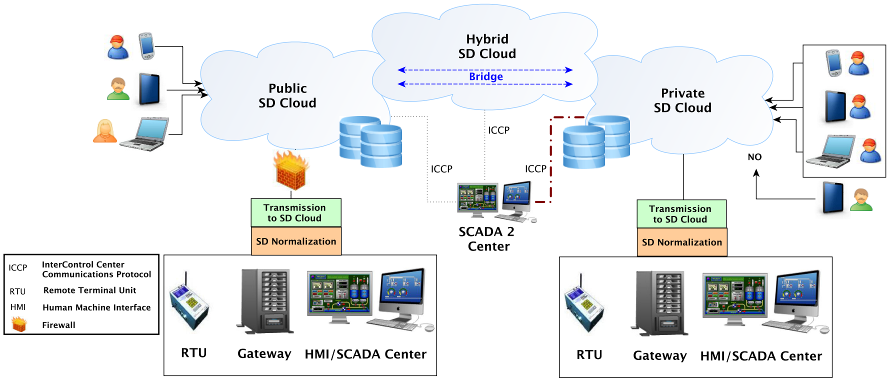

Figure 2: Cloud Computing for Critical Control Systems and WASA

The gateway is in charge of resending any type of information (0-5) from the WSN to the SCADA Center, interpreting and translating messages using GSAP (Gateway Service Access Point) points, and storing information copies in the SD cloud for backup. It is also responsible for anticipating future anomalies, managing critical alerts (4-5), and validating the entire model itself. For dealing with critical alerts, the gateway also has to locate the most suitable operator equipped with a hand-held device within the area. Regarding the SD cloud, we cannot ignore the fact that the environment is shared, where different providers and users can interact within the infrastructure via the Internet. In fact, there are two ways to integrate the SCADA services inside the cloud: either by executing services on-site where critical information is disseminated within the cloud using a hybrid/public cloud infrastructure (See Fig. 2, left hand side), or remotely executing them as part of the cloud through a private cloud infrastructure (See Fig. 2, right hand side). To protect critical data within the cloud, SD-based SCADA should be configured considering a private cloud infrastructure where the services offered by the service-oriented architecture should be operated solely by a single trustworthy organization. Virtualization of these services focuses on creating a virtual platform of hardware resources and operating systems to reduce costs, share information and manage resources from anywhere and at any time. In this way, we comply with WASA and the need to protect domains of the Grid over large geographic locations

3.1 Sensors and The Cluster Head for Dissemination and Detection

Fig. 3 depicts the chief modules of the CH: *Message Normalization*, *Pattern Association*, *Alarm Manager* (AM-CH), *Data Aggregation*, and *Diagnostics Manager*. Each sensor node (s_i) with identification IDs_i sends its messages (a v_i or an alarm) to its CH$_j$ with IDch_j, which first operates the Message Normalization module to combine and represent different data inputs in a generic format. The normalized message is then sent to the Pattern Association module in order to verify the nature of such inputs using simple behaviour patterns. For example, to verify whether readings or critical alarms received from a s_i are outside their acceptable thresholds before being forwarded to the gateway. In this way, we can make good use of CHs by supervising the functional instabilities of their nodes or the communication environment. These instabilities may be, for example, caused by malfunctions due to a lack of maintenance. Depending on the detected anomaly, the AM-CH module will generate, through the ARMO class, a new alarm signaled with high priority (4) so that an operator is made aware of the situation.

For the sake of simplicity, we consider that the network's deployment is based on small configurations of clusters where each sensor of the cluster has to transmit messages with the v_i and its priority, the IDs_i and the time-stamp. To address the malfunction problems, each CH must verify the payload of each message to check whether its v_i corresponds to the priority assigned by the sensor; e.g., $v_i \in$ (or $\notin$) $[V_{Low_{min}}, V_{Low_{max}}]$? Only in the case where a CH detects a discrepancy in the control by a sensor, does the CH have to penalize its behaviour by using four types of counters. These counters, unique to each node and initialized in the commissioning phase, represent the level of accuracy given in detection tasks. In particular, they are associated with four possible situations: (i) the sensor determines that an anomaly is occurring within the system, and it coincides with the CH (a True Positive (TP)); (ii) the sensor

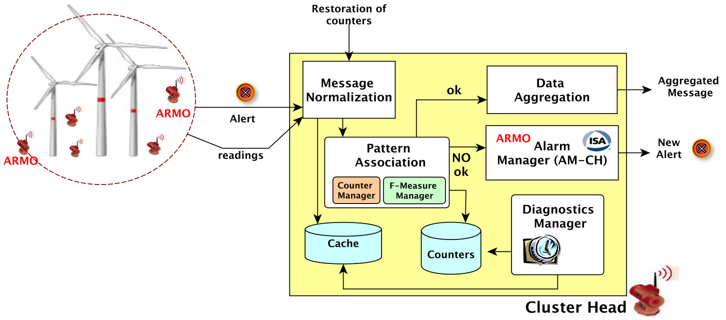

Figure 3: Architecture of the Cluster Head

determines that an anomaly is occurring within the infrastructure, and it does not coincide with the CH (a False Positive (FP); (iii) the sensor determines that no anomaly is occurring, and it does not coincide with the CH (a False Negative (FN)); and (iv) the sensor determines that no anomaly is occurring within the system, and it coincides with the CH (a True Negative (TN)). Depending on the behaviour of the sensor, one of these four counters has to be updated by one unit.

In order to carry out the accuracy tests, each CH also has to compute the *F-Measure* of each sensor to determine the level of precision in its observation tasks. F-Measure is a statistical measurement that corresponds to the harmonic mean of *precision* and *recall*, and which is weighted with a probabilistic value $\alpha \in [0, 1]$ [13]. In this context, α indicates the degree of precision in the detection tasks where a value $\alpha \simeq$ zero states a bad precision and a value $\alpha \simeq$ one represents a good precision. The technique of precision and recall, originally introduced for information retrieval [13], consists of calculating the ratio of correctly detected anomalies with respect to the rate of FP (Equation 1), as well as the ratio of correctly detected anomalies with respect to the rate of FN (Equation 2). Namely:

$$Precision = \frac{TP}{TP + FP} = \frac{correct\ warnings}{failure\ warnings} \in [0, 1] \tag{1}$$

$$Recall = \frac{TP}{TP + FN} = \frac{correct\ warnings}{real\ failures} \in [0, 1] \tag{2}$$

Given these two equations, F-Measure can then be computed as follows:

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \in [0, 1] \tag{3}$$

Given that the CH serves as a judge of the actions taken by a sensor, discrepancies of opinion can exist, and therefore a second judge (a human operator) with more objective decisions should also be taken into consideration. Hence, the system first trusts that the CH is a correct node in its evaluations and is able to increase the counters TP, FP and FN according to the situation, but this belief can change in the following phases. Specifically, the CH increases the counter values when: (i) the CH and the sensor do not coincide in the assigned priorities (a FP/FN), or when (ii) the situation detected by them is critical (a TP). The rest of the cases, which correspond to situations of consensus with priorities (0-3), are not managed by the CH since they are used as input for prevention and are therefore managed by the gateway. Only in the case where the second judge does not agree with the increase made by the CH, does the CH have to restore the counters involved, to their previous states. This cross-check for different elements of the WASAM is mainly due to the critical and delicate nature of the controlled infrastructures, which obligates the system to configure hybrid solutions based on human objections [4]. Note that this type of actor does not focus exclusively on assisting in situations where there are discrepancies between priorities. The human operator is also responsible for attending to those critical scenarios (4-5) where the CH and the sensor reach a consensual critical priority. Given the relevance of the role of this actor, this is described in more detail below.

On the other hand, hardware problems are managed using the Diagnostics Manager, which periodically queries the last sequence of events received from the sensors using a cache memory. This memory, which is maintained by the Message Normalization, allows the Diagnostics Manager to know when a particular node of the cluster is not sending messages for a short time period. If this occurs, the CH infers that something anomalous is happening with the sensor, and updates its counter of FN by one unit. This problem could be attributed to interferences in the communication, a significant reduction in battery levels or possibly the lifetime of the sensor is over and impedes a sensor in coordinating its observation tasks properly. It should be noted that this updated value of FN coincides with the counter FN assigned to each node, because when a node is behaving incorrectly, the system increases (without any distinction of the cause) its value until its value of F-Measure reaches the $Threshold_{min}$ ($\simeq$ zero). At that point, the CH will have to warn of the situation so that the environment of it can be checked, and the sensor can be tested or discarded.

To generate a new alarm, both the Pattern Association and the Diagnostics manager will have to send the AM-CH a data set. For example, the IDch_j, IDs_i, the type of alarm (only if the received message from s_i is an alarm), the priority assigned by the sensor, the priority assigned by the CH, and the type of event detected. The kind of event is an indicator that will help make the gateway and the human operator aware of the type of problem to be checked. In particular, two types of events are used: *event_detSensor* and *event_detCH*. The former refers to the detection made by a sensor node (i.e., the control of the critical infrastructure and its services), whereas the latter is attributed to the detection carried out by the CH (i.e., the control of conduct within the cluster). To show the simplicity of the Pattern Association module, Algorithm 3.1 summarizes the order of execution of its actions.

Algorithm 3.1: CLUSTER HEAD()

```
local $IDs_i, message, v_i, prCH, prSensor, fmeasure$;

while RECEIVINGMESSAGES()
  do
    $message \leftarrow$ NORMALIZEDMESSAGE();
    $v_i \leftarrow$ EXTRACT_READINGDATA($message$);
    $\boldsymbol{prSensor} \leftarrow$ EXTRACT_PRIORITY($message$);
    $IDs_i \leftarrow$ EXTRACT_IDENTIFIERSENSOR($message$);
    $\boldsymbol{prCH} \leftarrow$ DETERMINEPRIORITY($v_i$);

    if CONSENSUS($\boldsymbol{prCH}, \boldsymbol{prSensor}$)
      then
        if ($\boldsymbol{prSensor} == Journal$)
          then {DATA_AGGREGATION($IDs_i, v_i$);
          else
            FORWARDALARM($IDch_j, IDs_i, v_i, prSensor$, '$event\_detSensor'$);
            if ($\boldsymbol{prSensor} >= High$)
              INCREASE_TP($IDs_i$);
              $fmeasure \leftarrow$ CALCULATECURRENT_FMEASURE($IDs_i$);
      else
        if ($\boldsymbol{prCH} < \boldsymbol{prSensor}$)
          then
        {INCREASE_FP($IDs_i$);
          else {INCREASE_FN($IDs_i$);
        $fmeasure \leftarrow$ CALCULATECURRENT_FMEASURE($IDs_i$);
        if $fmeasure \leq Threshold_{min}$
          then {$NewAlarm(IDch_j, IDs_i, high$, '$event\_discardNode'$);
          else {$NewAlarm(IDch_j, IDs_i, high, \boldsymbol{prSensor}, \boldsymbol{prCH}$, '$event\_detCH'$);
```

3.2 A Powerful Gateway for Prevention, Response and Maintenance

The gateway is composed of two chief managers: An *Incident Manager* and a *Maintenance Manager* (See Fig. 4).

3.2.1 Incident Manager: Prevention, Data Redundancy and Response.

Any type of information received from CHs is taken through the *ARO* sub-module, which temporarily stores them within a cache memory and sends a copy to both the SCADA Center and the SD cloud. For incident management, ARO uses one organized queue, which is sorted by priorities. Depending on the criticality of the message, the *Alarm Manager* (AM-GW) sub-module will carry out two actions; one predictive and the other reactive. For the predictive part, the AM-GW must compute the rate of valid readings (0) and non-critical alarms (1-3) received from the network. The idea is to calculate, for each sensor, the rates of consecutive values of non-critical alarms with value 3 over the last time period, as it may mean the proximity of a possible incident. Therefore, we propose here (See below) a simple prevention method included inside the *Prediction* sub-module belonging to the AM-GW. This means that the predictive module is not exclusive and other forecast schemes [13] can be added to the architecture if needed.

The method consists of calculating probabilities of transition between states: st_0 (represents valid readings), st_1, st_2, st_3 (represent different types of criticality

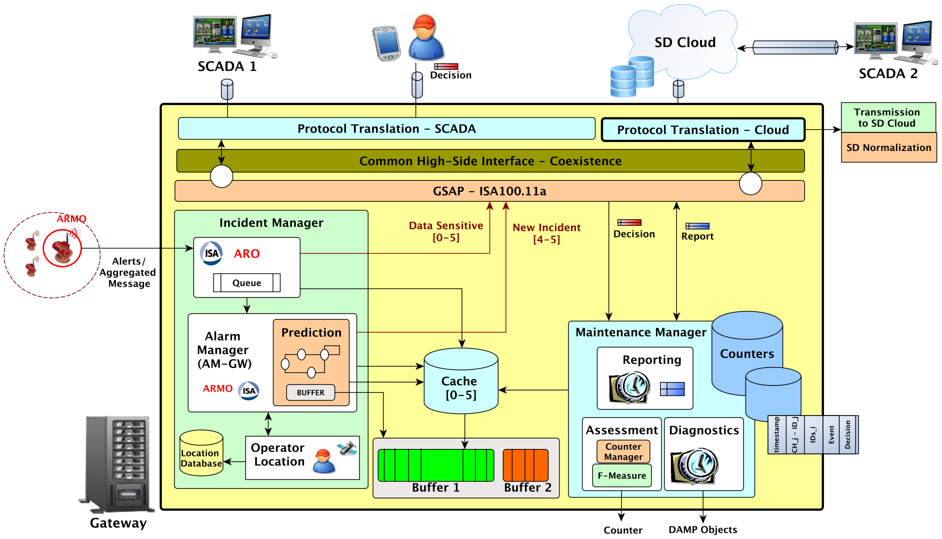

Figure 4: Architecture of the ISA100.11a Gateway

(1-3)). These states and their values have to be previously exported from the cache memory to a separate temporal buffer, which is assigned to each network sensor, Bff_i, with a size Δ_{Bff_i}. However, this buffer is not only based on information exported from the cache, but also on a small percentage of past information in order to keep a sequence of events with respect to the time line. Therefore, the size of Bff_i is based on exported information with a size of Δ_{Bff1_i} and on past information with a size of Δ_{Bff2_i}; i.e., $\Delta_{Bff} = \Delta_{Bff1} + \Delta_{Bff2}$, where $\Delta_{Bff1} \geq \Delta_{Bff2}$. In this way, we can restrict the size of Δ_{Bff} and reduce computational costs by not computing the predictive algorithm and the cache several times.

For each of the states, we also design a particular probability of transition pr_{st_α,st_β}, which corresponds to the probability of going from a state α to a state β; i.e., we define for prevention $pr_{st_\alpha,st_\beta} = Pr(st_{i+1} = \beta | st_i = \alpha)$, where $\sum_{i=0}^{3} pr_{st_\alpha,st_\beta} = 1$. Taking this into account, we assume that the probability (pr_{st_i}) of remaining in a *normal situation* st_0 (i.e., a state without containing a priority situation/alarm) is much greater than transiting to an *unstable state* st_3 or remaining within this; i.e., $pr_{st_0} > pr_{st_1} > pr_{st_2} > pr_{st_3}$. In this way, we are able to represent, with high probabilities, the fact of remaining at normal states with respect to those states with criticality. In a nutshell, if an anomaly is occurring or is starting to appear within the system, the method will return low transition values to warn of the situation.

In order to calculate these probabilities, we consider as an initial approach the following Equation: $1/(4 \times \alpha)$, where α represents the non-critical current state, such that $\alpha <= 3$ (st_3) and $\alpha > 0$ (st_0) since $pr_{st_0} = 1 - (\sum_{\alpha=1}^{3} pr_{st_\alpha})$.

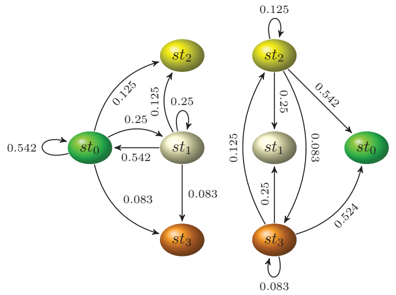

Figure 5: An Example of Transition between States: From st_α to st_β

Note that other approaches could be equally valid if they comply with the restriction of $pr_{st_0} > pr_{st_1} > pr_{st_2} > pr_{st_3}$. The result of computing the probabilities for each state is illustrated in Fig. 5 where relationships between states together with the cost of their transitions are also depicted. On the other hand, and considering the previous assumptions and notions, the occurrence of an event can be computed as follows.

$$\frac{InitialState + \sum_{j=0}^{\Delta_{Bff_i}-1} pr_{Bff_i[j],Bff_i[j+1]}}{\Delta_{Bff_i}} \leq (pr_{st_3} + \sigma_{error}) \tag{4}$$

where *InitialState* corresponds to $pr_{Bff_i[0]}$ and σ_{error} shows an acceptable margin of error. The interpretation of this equation is twofold. On the one hand, if the result of computing Equation 4 is $\leq pr_{st_3} + \sigma_{error}$, the system can determine that the next value to be received will be either a non-critical alarm with value 3 or a critical alarm. On the other hand, if the result of computing Equation 4 is $> pr_{st_3} + \sigma_{error}$, it may infer that the next entry may be either a v_i or a non-critical alarm. For consecutive values with a medium value (e.g., 3 3 3 3 3 3 3 3 3 3) require both the SCADA Centre and the nearest operator within the affected area being alerted, through the AM-GW. For operator location, the AM-GW uses the *Operator Location* sub-module, which considers the operator's availability according to their contract, and the use of geospatial positioning devices to locate their position within the affected area. Lastly, and as mentioned in Section 2, the AM-GW has to send a copy of the incident both to the SCADA Center and to the SD cloud using a *SD Normalization and Transmission* module to standardise the dissemination within the cloud (See Fig. 2 and Fig. 4).

3.2.2 Maintenance Manager: Self-validation and Maintenance.

In order to know the real state of the entire model, the *Assessment* sub-module needs to receive certain feedback (denoted as *prOp*) on how accurate the prevention and detection modules have been. This feedback is dependent on the final decision of operator, who is obliged to validate the degree of reliability, which

	Prevention	**Detection and Control of the CI**	**Detection and Control of the Cluster**					
	priority	*prSensor*	*prCH*			*prSensor*		
priorityOp.	**High**	**High**	**0**	**(1-3)**	**(4-5)**	**0**	**(1-3)**	**(4-5)**
Normal Sit.	FP	FP	TP	FP	FP	TP	FP	FP
Unstable Sit.	FP	FP	FN	TP	FP	FN	TP	FP
Critical Sit.	TP	TP	FN	FN	TP	FN	FN	TP

Table 1: Behaviour Assessment for Prevention, Detection and Control

is done through their hand-held interfaces. The operators' decision (*prOp*) depends on several ISA100.11a criticality levels, categorized as: *normal situation*, *unstable situation*, and *critical situation* (cf. Section 3 and Table 1). Depending on the level of criticality taken by the operator, the system has to update values of TP, FP, TN, FN for the respective modules of the system using Equation 3. Only in the case where the value of F-Measure reaches its $Threshold_{min}$, does the Assessment sub-module have to issue a new alarm with a high priority, through the AM-GW. The new alarm should contain information related to the nodes involved (e.g., IDs_i, IDch_j, IDgw) and the action to be carried out, such as *event_review_detModule*, *event_review_predModule*, or even *event_discardNode*.

For the F-Measure evaluation of the prediction module, it is enough to take into account the operator's decision and the estimation made by the Prevention sub-module. If the operator's feedback notes that the current situation is under threat or it is really critical, the prediction sub-module should then be rewarded in some way (increasing TP by one unit), otherwise it should be penalized accordingly (increasing FP by one unit). Similarly, this assessment method can be applied to evaluate the reliability of sensors in their observation tasks, and the reliability of CHs in their supervision tasks. In particular, the method can be applied, taking into account two possibilities: (i) The $prCH$ of the CH$_j$ coincides with the $prSensor$ of the s_i (i.e., consensus of a critical situation (4-5)); and (ii) the $prCH$ does not coincide with the $prSensor$ (i.e., discrepancies between priorities (0-5)). For the former, a further two situations may arise:

- The *prOp* coincides with both the $prCH$ and the $prSensor$; i.e., a TP in both CH$_j$ and s_i. In this context, the operator's feedback indicates the existence of a critical situation (4-5) and the system should reward such effectiveness by increasing the counter TP of the CH by one unit. Note that the TP of the sensor was already considered in Section 3.1.

- The *prOp* does not coincide with either the $prCH$ or the $prSensor$; i.e., a FP in both CH$_j$ and s_i. The operator's feedback does not indicate the existence of a critical situation. In this case, the system should, on the one hand, increase the FP of the CH; and on the other hand, penalize the sensor by restoring its TP and increasing its FP.

Algorithm 3.2: ARBITRAGE OF PRIORITIES($prCH, ID_{CH_j}, prSensor, ID_{s_i}, prOp$)

```
if CONSENSUS($prOp, prCH, prSensor$)
  then
    {INCREASE_TP($ID_{CH_j}$);
  else {
    if CONSENSUS($prOp, prCH$))
      then
        {INCREASE_TP($ID_{CH_j}$);
      else {
        if ($prOp > prCH$)
          then
            {INCREASE_FN($ID_{CH_j}$);
          else {INCREASE_FP($ID_{CH_j}$);
        if CONSENSUS($prOp, prSensor$))
          then
            {RESTORE_FN/FP($prCH, ID_{CH_j}, prSensor, ID_{s_i}$);
             INCREASE_TP($IID_{s_i}$);
          else {RESTORE_FN/FP($prCH, ID_{CH_j}, prSensor, ID_{s_i}$);
                INCREASE_FN/FP($IID_{s_i}, prOp, prSensor$);
```

However, the latter method (non-consensual priorities) is a little more complex because of the need to compare versions with the criticality provided by the operator (See Algorithm 3.2). When comparing versions, a further two specific situations may take place:

- The *prOp* coincides with the *prCH*; i.e., a TP in CH. The system rewards the CH by increasing its TP by one unit, and maintains the penalization of the sensor (cf. Section 3.1). It is possible to think here that a sensor can be unfairly penalized when the physical layer (noise/interferences) is the main reason for the perturbation in the messages. However, this observation is also managed by the WASAM model by controlling the rates of FP/FN related to a sensor and indirectly to its environment (e.g., industrial noise).

- The *prOp* does not coincide with the *prCH*; i.e., a FP/FN in CH. The system increases the FP/FN of the CH accordingly. However, a further two cases may also occur when the sensor needs to be evaluated. On the one hand, the *prOp* coincides with the *prSensor* (i.e., a TP in s_i). This means that the system needs to reward the s_i by restoring its FP/FN and increasing its TP (cf. Section 3.1). On the other hand, if the *prOp* does not coincide with the *prSensor* (i.e., a FP/FN in s_i) the system penalizes the behaviour of the sensor s_i according to the *prOp*. However, this penalization also relies on the penalization carried out by the CH in Section 3.1. For example, if $prOp$ (0) $<$ $prSensor$ (1-3), but the $prSensor$ (1-3) $<$ $prCH$ (4-5) and $prOp$ (0) $<$ $prCH$ (4-5), then s_i should restore its FN and increase its FP accordingly.

In order to extend the functionality of the model, a *Diagnostics Manager* is also used to check the lifetime of the CHs. As the Diagnostics Manager of Section 3.1 does, it will have to frequently check whether a specific CH$_j$ stopped sending messages during a significant time period by analysing its sending frequency in the cache memory. If this occurs, the manager will have to diagnose its current state by sending a message based on DMAP objects. If the CH_j does not respond within a maximum time limit, the manager will have to warn of the situation using the type of event *event_discardCH*. These diagnoses allow the system to manage isolated areas caused by malfunctions or denial of service attacks in CHs, given that these nodes are normally a single failure points. Obviously, this action should be carried out for each network node, but this could mean a degradation of performance. For this reason, we supervise the lifetime of sensors using their counter FN, and thus we avoid an increase in the communication overhead. Finally, the system should allow the SCADA Center to periodically or on-demand receive reports with accumulative values of F-Measure through the *Reporter* sub-module so as to know the real situation of the system.

Given that critical data stored within a cloud can be computed without a priori known location, protection and privacy aspects are required. This protection includes the use of cryptographic services during the upload/download phase of information within the cloud and during its storage phase. Some of these services have already been identified and analysed in [3] for Smart Grid domains, such as: Searchable Encryption protocols with the capability to search over encrypted data, enabling a server to execute queries without having to decrypt the data; digital signature schemes to encrypt atomic data; or the use of Proofs of Storage protocols to check the integrity of large amounts of data stored in a server without needing to retrieve it. Privacy issues can also be addressed by applying Searchable Encryption protocols and Private Information Retrieval protocols to retrieve information from a database without revealing any information about the requested data to the cloud server, or through Anonymous Routing protocols to ensure anonymity during online communications. In addition, it is also essential to protect the visibility of virtualization of resources, balance the standardized backup copies within the cloud, and monitor any activity within the cloud in a regulated way [3, 14]. On the other hand, although security aspects are beyond the scope of this paper, we assume that communication channels ‘sensor-sensor’ are protected by using security credentials and cryptographic services provided by the ISA100.11a standard [7]. The rest of the communications will depend on security services of the TCP/IP standard, virtual private networks with the Internet Protocol Security (IPSec) tunnel mode, as well as diode/unidirectional communication, firewalls and intrusion detection systems (See Fig. 1).

4 Simulations and Discussions

The implementation is based on a small scenario composed of two virtual clusters with two sensors each and three virtual operators with random actions and different work availabilities. In order to study the worst cases and analyse the behaviour of the entire model, we have intentionally stressed the context to randomly produce and attend FPs, FNs and TPs[1], and thus obtain diverse

[1] Although the counter TN has no place in this analysis in particular, it could be considered for real applications.

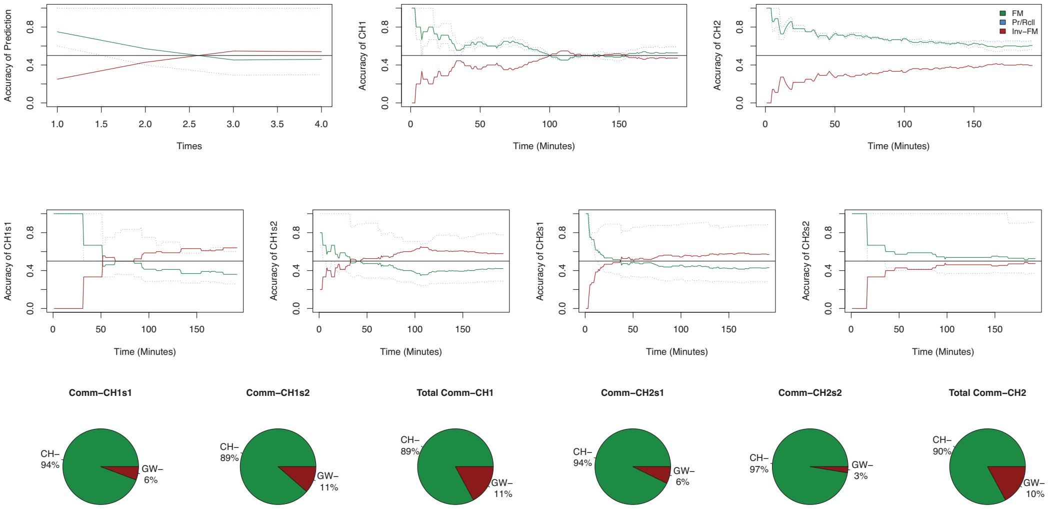

Figure 6: Test of Accuracy using F-Measure Together with the Communication Overhead

results for the analysis. The part of the sensor network defined in Section 3.1 has been implemented in nesC and simulated through the Avrora simulator under the de-facto standard operating system for sensor nodes, TinyOS 2.x. Avrora is able to interpret conventional sensor nodes (e.g., Mica2) belonging to the category II defined in [10]; i.e., 4-8 MHz, 4-10 KB RAM, 48-128 KB ROM with 2-8 mA of energy. The results of the simulation indicate that a cluster working as a Mica2, requires less than 8 MHz to execute the software, consuming around 3,3 Joule for CPU and 8.6 Joule for radio, and approximately reaching a maximum of 2.8% for reading and a 3% for writing in memory. Therefore, if traditional sensors are able to work as CHs, then ISA100.11a sensors belonging to category III with higher capabilities are also able to serve as CHs. On the other hand, the techniques integrated inside the gateway (See Section 3.2) have been implemented in Java.

In this light and considering that each model component (i.e., sensors, CHs and the prevention module) is initialized with counters of TP, FP, TN and FN at zero, the results of the simulation are illustrated in Fig. 6[2], where the relevance of the F-Measure technique (cf. Section 3.1) for the behaviour assessment is clearly depicted with respect to the time line. In addition, Fig. 6 also indicates the influence of FN/FP on the reliability of the system and its components since both parameters significantly affect the test of accuracy as happens in CH1s1, CH1s2 and CH2s1. Hence, thanks to this test it is possible to identify those situations which can become unacceptable in critical environments where existing

[2] FM represents the value of F-Measure computed for each sensor at each time, Pr/Rcll depict the ratios associated to the precision and recall, and Inv-FM represents the mirror of the F-Measure to intensify the signal from the worst situation; $\alpha \simeq$ zero.

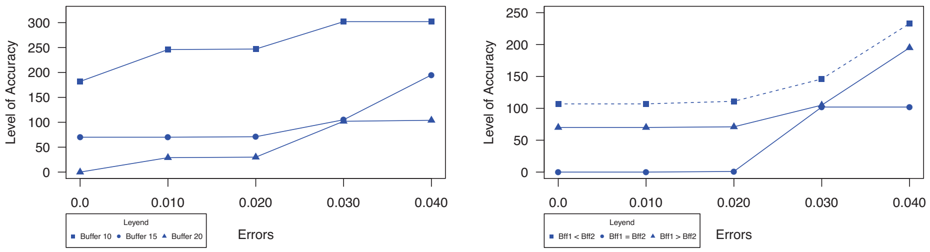

Figure 7: Left Hand Side Figure: The Importance of σ_{error} and Δ_{Bff} for Critical Contexts; Right Hand Side Figure: The Importance of Δ_{Bff1} and Δ_{Bff2}

anomalies are not detected properly. Moreover, Fig. 6 also illustrates the cost of communication between cluster heads and the gateway, where the CH1 has reduced its communication in 89% and the CH2 in 90%. Therefore where the rate of FP/FN/TP of sensors were kept by the CH but entirely managed by the gateway, the communication cost could become more significant.

It is also important to define a suitable value for σ_{error} and an appropriate buffer size Δ_{Bff} (See Section 3.2.1) for prevention. The higher the margin of error and the smaller the buffer is, the greater the probability of obtaining a high number of FPs; i.e., the level of accuracy represented in Fig. 7 through the vertical axes. In particular, this figure shows sequences of events (intentionally stressed from the previous simulations) analysed according to different sizes of Δ_{Bff1} (5, 10, 15) and Δ_{Bff2} with value 5, as well as different values of σ_{error} (0.0, 0.010, 0.020, 0.030 and 0.040). Given this, Δ_{Bff} then takes the following values 10, 15, 20 since $\Delta_{Bff} = \Delta_{Bff1} + \Delta_{Bff2}$. The analyses indicate, on the one hand, that a system configured with a Δ_{Bff} size of 10 is less restrictive and precise than using a buffer with a size of 20 (See Fig. 7 (left hand side)). This is also the case when the system is configured with a σ_{error} with value of 0.040. On the other hand, Fig. 7 (right hand side) represents the importance of determining the sizes of Δ_{Bff1} and Δ_{Bff2}. The results indicate that a $\Delta_{Bff1} \geq \Delta_{Bff2}$ (continued line – $\Delta_{Bff1} = 10$ and $\Delta_{Bff2} = 5$; and $\Delta_{Bff1} = 10$ and $\Delta_{Bff2} = 10$) is more precise than using a $\Delta_{Bff1} < \Delta_{Bff2}$ (dashed line – $\Delta_{Bff1} = 5$ and $\Delta_{Bff2} = 10$). The reason being that the system can be able to compare more current information with a small portion of past information, and this way to track the behaviour of the sensors in that time period.

Finally, it is advisable to have good software maintenance of sensors, as their output is the input of the prevention. This can be seen as a dependency relationship of 'cause-effect'. If a sensor does not work properly, the prediction then has a tendency for FP/FN. Therefore, the role of the CH for detecting malfunctions and the role of the Maintenance Manager for controlling anomalous conduct in the entire system are essential to avoid instabilities in the final prediction. Lastly, note that although WASAM has been designed for Smart Grid environments, it can be extrapolated to other critical contexts such as transport systems or water treatment systems.

5 Conclusions

A dynamic wide-area situational awareness model based on the composition of different technologies has been proposed in this paper to provide a set of benefits for control systems. The purpose is to offer an effective support to anticipate unforeseen situations before disruptions can arise within the observed infrastructure, or even within the observation system from anywhere and at any time, in addition to offering reactive capacities when such situations definitively appear within the system. Nonetheless, more work on situational awareness is still needed. It is necessary to explore new technologies and techniques to adapt them to the context without compromising the security and performance of the underlying system. For example, for the worst cases, we highlight aspects of controllability to allow an affected system to reach its normal configurations in a desired and finite set of steps.

Acknowledgments

This work has been partially supported by the research projects ARES (CSD2007-00004), PISCIS (P10-TIC-06334), the EU FP7 project FACIES (HOME/2011/CIPS/AG/4000002115). The first author also receives research funding from the Marie Curie COFUND programme "U-Mobility" co-financed by the Univ. of Malaga and the EC FP7 under GA No. 246550.

References

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

[2] E. Knapp (2011), *Industrial Network Security, Securing Critical Infrastructure Networks for Smart Grid SCADA, and other Industrial Control Systems*, Elsevier, pp. 1-360.

[3] C. Alcaraz, I. Agudo, D. Nunez, and J. Lopez (2011), *Managing Incidents in Smart Grids à la Cloud*, In IEEE CloudCom, pp. 527-531.

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

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

[6] A. Mavridou, M. Papa (2012), *A Situational Awareness Architecture for the Smart Grid*, ICGS3/e-Democracy, Social Informatics and Telecommunications Engineering, Springer, LNCS 99, pp. 229-236

[7] ISA100.11a (2009-2013), *ISA-100.11a-2009. Wireless Systems for Industrial Automation: Process Control and Related Applications*, The International Society of Automation, Retrieved on Dec. 2012.

[8] A. Mohsenian-Rad, and A. Leon-Garcia (2010), *Coordination of Cloud Computing and Smart Power Grids*, First IEEE Smart Grid Communications, pp. 368-372.

[9] CloudCERT (2012-2014), *Testbed Framework to Exercise Critical Infrastructure Protection*, European Project, HOME/2010/CIPS/AG/20.

[10] J. Lopez, R. Roman, and C. Alcaraz (2009), *Analysis of Security Threats, Requirements, Technologies and Standards in Wireless Sensor Networks*, FOSAD 2009, Springer LNCS 5705, pp. 289-338.

[11] C. Alcaraz, and J. Lopez (2012), *Analysis of Requirements for Critical Control Systems*, IJCIP, Elsevier, vol. 2, no. 3-4, pp. 137-145.

[12] *Anomalous Situation* (2012), `\http://oxforddictionaries.com/definition/anomalous`, Oxford Dictionary, Retrieved on Dec. 2012.

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

[14] I. Abbadi (2013), *A Framework for Establishing Trust in Cloud Provenance*, International Journal of Information Security, vol. 12, no. 2, pp. 111-128.