The ICT monitoring system of the ASTRI SST-2M prototype proposed for the Cherenkov Telescope Array

Gianotti, F., Bruno, P., Tacchini, A., Conforti, V., Fioretti, V., et al.

F. Gianotti, P. Bruno, A. Tacchini, V. Conforti, V. Fioretti, C. Tanci, A. Grillo, G. Leto, G. Malaguti, M. Trifoglio, "The ICT monitoring system of the ASTRI SST-2M prototype proposed for the Cherenkov Telescope Array," Proc. SPIE 9913, Software and Cyberinfrastructure for Astronomy IV, 99132I (8 August 2016); doi: 10.1117/12.2231609

SPIE.

Event: SPIE Astronomical Telescopes + Instrumentation, 2016, Edinburgh, United Kingdom

The ICT monitoring system of the ASTRI SST-2M prototype proposed for the Cherenkov Telescope Array

F. Gianotti (*)(a), P. Bruno (b), A. Tacchini (a), V. Conforti (a), V. Fioretti (a), C. Tanci (c), A. Grillo(b), G. Leto (b), G. Malaguti(a), M. Trifoglio(a), for the ASTRI Collaboration(d) and the CTA Consortium (e)

(a) INAF - Istituto di Astrofisica Spaziale e Fisica Cosmica di Bologna, Via P. Gobetti 101, 40129 Bologna, Italy

(b) INAF – Osservatorio Astrofisico di Catania, Via S. Sofia 78, 95123 Catania, Italy

(c) INAF - Osservatorio Astronomico di Brera, Via Brera 28, 20121 Milano Italy

(d) http://www.brera.inaf.it/astri/

(e) http://www.cta-observatory.org/

(*)gianotti@iasfbo.inaf.it

ABSTRACT

In the framework of the international Cherenkov Telescope Array (CTA) observatory, the Italian National Institute for Astrophysics (INAF) has developed a dual-mirror, small-sized, telescope prototype (ASTRI SST-2M), installed in Italy at the INAF observing station located at Serra La Nave, Mt.Etna. The ASTRI SST-2M prototype is the basis of the ASTRI telescopes that will form the mini-array proposed to be installed at the CTA southern site during its pre-production phase. This contribution presents the solutions implemented to realize the monitoring system for the Information and Communication Technology (ICT) infrastructure of the ASTRI SST-2M prototype.

The ASTRI ICT monitoring system, has been implemented by integrating traditional tools, used in computer centres, with specific custom tools which interface via Open Platform Communication Unified Architecture (OPC UA) to the Alma Common Software (ACS) that is used to operate the ASTRI SST-2M prototype. The traditional monitoring tools are based on Simple Network Management Protocol (SNMP) and commercial solutions and features embedded in the devices themselves. They generate alerts by E-mail and SMS. The specific custom tools convert the SNMP protocol into the OPC UA protocol and implement an OPC UA server. The server interacts with an OPC UA client implemented in an ACS component that, through the ACS Notification Channel, sends monitor data and alerts to the central console of the ASTRI SST-2M prototype. The same approach has been proposed also for the monitoring of the CTA on-site ICT infrastructures.

Keywords: very high energy gamma ray astronomy, CTA, Small Size Telescope, Telescope prototyping, ASTRI, Telescope System Software, Information and Communications Technology, ICT, Monitoring, ACS, OPC-UA, SNMP

1. INTRODUCTION

ASTRI (Astrofisica con Specchi a Tecnologia Replicante Italiana) is a project of the Italian National Institute for Astrophysics (INAF) funded by the Italian Ministry of Education, University and Research (MIUR) and closely linked to the development of the Cherenkov Telescope Array (CTA[1]) Observatory. Within this framework, INAF is currently developing an end-to-end prototype of the CTA small-size telescope in a dual-mirror configuration, named ASTRI SST-2M[2], to be tested under field conditions and scheduled to start data acquisition in 2016.

A further objective of the ASTRI program is the definition, development, and deployment, at the CTA Southern site, of a mini-array of at least nine ASTRI telescopes which is proposed to constitute the first seed of the CTA Observatory. For that reason the ASTRI SST-2M prototype design already takes into account all the hardware and software prerequisites that are necessary in view of the future ASTRI mini-array[3]. Among them, the necessary Interface Communications Technology (ICT)[15] equipment of the prototype has been designed such that it might be scaled for the future evolution in the mini-array configuration and then in compliance with the specifications provided in the CTA framework. The following requirements have been taken carefully into account:

- the Cherenkov telescopes acquire a huge amount of data (from 100TB of ASTRI to 10PB of CTAper year);
- the data reduction is complex and should ensure the on-line analysis of the data flow with no backlogs[16].
- the telescopes will be placed in remote and isolated sites;
- a large processing capability is needed on site, enough to process the large amount of data acquired.

The ICT equipment shall support the whole use case scenarios foreseen for the ASTRI telescope[14].

Software and Cyberinfrastructure for Astronomy IV, edited by Gianluca Chiozzi, Juan C. Guzman, Proc. of SPIE Vol. 9913, 99132I · © 2016 SPIE
CCC code: 0277-786X/16/$18 · doi: 10.1117/12.2231609

The ASTRI SST-2M prototype is installed in Italy at the INAF observing station located in Serra La Nave, 1735 m a.s.l. on Mount Etna, Sicily. The Serra La Nave (SLN) station is connected to the Internet within the INAF network by means of a direct radio link. A stand-alone ICT infrastructure is implemented at the SLN[5] site as part of the telescope prototype; it includes computer servers and workstations, network devices, an uninterruptable power supply system, air conditioning systems and auxiliary instrumentation devoted to the control or monitoring of environmental parameters. Suitable hardware and software tools will allow the parameters related to the behaviour and health of each item of equipment to be controlled and monitored. The ASTRI SST-2M ONSITE monitoring will consist of two main independent monitoring systems running in parallel (see Figure 1):

- a traditional monitoring system for the computer centre, based on an SNMP[6] framework.
- a monitoring system developed for ACS, making use of SNMP to OPC-UA[7] conversions.

In addition, the ASTRI SST-2M ONSITE monitoring may use in special cases specific monitoring tools embedded in the equipment e.g., Intelligent Platform Management Interface (IPMI), Web based tools, command line tools.
The design of the monitoring system in the ASTRI-SST-2M framework is the purpose of this contribution. Sections 2-3 give a general overview of the proposed architecture and technical solutions that integrate the ICT equipment. Section 4 describes the traditional monitoring system for the computer centre. Section 5 describes instead how the monitoring systems are integrated in the Observatory Control System package of the ASTRI Mini Array Software System (MASS[4]), while Section 6 describes in detail a specific end-to-end test case in the framework of the ASTRI SST-2M prototype.

2. DESIGN CONCEPT

The SNMP[6] protocol is a de facto standard for monitoring applications and it is supported by most of the devices on the market. For these reasons both monitoring systems of the ASTRI project are based on SNMP.
The SNMP is a component of the Internet Protocol Suite as defined by the Internet Engineering Task Force and it consists of a set of standards for managing Internet Protocol (IP) devices (e.g. routers, switches, servers, workstations, printers, modem racks), providing its users with a set of operations that allows these devices to be managed remotely.
A monitoring system using SNMP can retrieve information from any device supporting the protocol. This allows us to develop a monitoring system which is almost independent from the hardware and to exploit the available SNMP-based contributed software and standard tools.

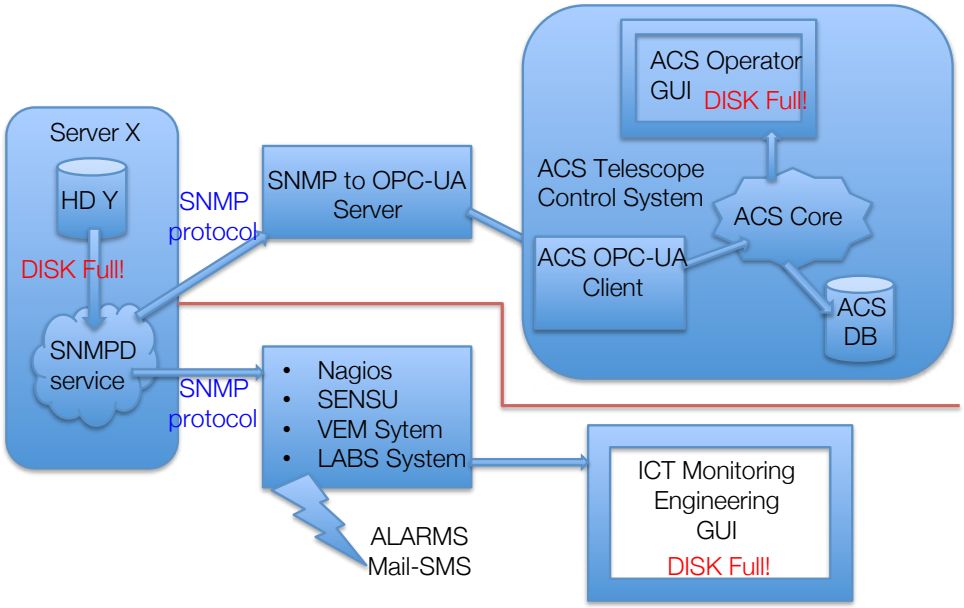

Figure 1 Sample of how the two monitoring systems operate in parallel from a common interface: DISK Full Alarm

The design goal for the traditional monitoring system is to build a set of SNMP managers that allows accessing all the parameters of interest by organizing them in a simple way in a single user-friendly GUI. This system is mainly dedicated to ICT managers, but it can also provide useful information to ordinary users.

Since the ICT monitoring management should also be integrated in the overall MASS software based on the ACS framework, we will build a set of dedicated SNMP to OPC-UA servers to convert a subset of the SNMP parameters coming from the ICT. In this way the ICT monitoring parameters will be available in the ACS system of the MASS software and may be stored in the database and / or be displayed in the operator GUI. The two systems, the traditional SNMP-based system and its subset interfaced to the ACS framework, will run in parallel.

3. EQUIPMENT AND PARAMETERS TO BE MONITORED VS SNMP

3.1 Equipment to be monitored

Now that we have defined the SNMP interface from which to start the monitoring systems we have to decide what to monitor. We define node as each component that has to be monitored on the network. Different classes and sub-classes of nodes are identified:

• **Network Connections**	• **Network Devices:** o Central switches. o Telescope switches. o Management switches. o Access point WiFi.
• **Infrastructure Devices :** o UPS. o Conditioner. o Power generators. o Sensors eg. temperature, humidity etc…	• **Servers/Workstation**: o Computing servers. o Services servers. o Storage servers. o Telescope control servers. o DAQ camera servers.

For each of these classes we will have to decide which are the parameters to be monitored and what are the limits in which these parameters are nominal or are considered in error by the out-of limit check. It's very important to consider only the most important parameters that can automatically and surely detect a problem.

3.2 Parameters to be monitored

The first crucial stage is monitoring, through the Internet Control Message Protocol (ICMP) protocol, the **UP / DOWN status** of every node in the network.

The **Network Connections** represents the simplest node class, for this class of node it is only possible to monitor the status and network traffic. For the **Infrastructure Devices** node class we can monitor only the *hardware* or *physical parameters* (e.g., temperature FAN velocity, current, voltage, power levels, etc.). For the **Network Devices** node class we must monitor *hardware* or *physical parameters* and *system parameters.* The system parameters are related to the functional operation of the device and for the proper functioning of its Firmware or Operating System. System parameters to be monitored are typically CPU load, memory use etc.

The most complex node class is the **Server/Workstation** since it has three categories of parameters to be monitored: (Figure 2): *hardware* or *physical parameters, system parameters,* and *software parameters*. The software parameters include all the monitor information about the operation of: daemons, services and applications.

The ICT hardware choices strongly influence the potential of monitoring. The physical parameters are related to the kind of hardware used and the probes that are implemented. It will be necessary to determine the common set of parameters that are present in all the equipment of a class. Something similar must be made to the system parameters because when a parameter is not expected to be monitored, we cannot know its behaviour. Anyway devices in Infrastructure Devices and Network Devices classes are not user programmable so we will study their Management Information Base (MIB) to understand what are the parameters we can monitor. Only for the Server/Workstation class can we configure the SNMP daemon and also by adding standard plug-in so we can modify the MIB adding some parameters to be monitored.

Server			Network		
		description			description
Hardware	Memory	transition to Non-Critical from OK Correctable memory error	Hardware	Memory	
	Temperature	Lower Non-critical - going low Upper Non-critical - going high		Temperature	
	PS Redundancy	Fully Redundant => Redundancy Lost		CPU	
	Voltage	State Deasserted => State Asserted		Power Supply	
	Power Supply	Fully Redundant => Redundancy Lost Power Supply Failure detected Power Supply input lost (AC/DC)		Fan	
	Fan	Fully Redundant => Redundancy Lost Lower Critical - going low		Line card fail supervisor fail fabric fail	trap
	Entity Presence	Entity Present => Entity Absent		Failover	trap
	Disks	Media/Predictive Errors, Failures		port fail	trap (error, link status-->telescopes & trunk connections)
	firmware	Version			
	RAID Controller	RAID Volume Integrity			
	CPU	Erorr			
System	Bonding		System	routing change	ospf
	CPU	Clock Rate, Load:: Threshold, to be adapted per machine		spanning tree change	trap
	Memory	Speed+Usage		configuration change	trap
	Filesystems	functionality			
	Prozesses	zombies, abnormal stuff			
	Network	Bandwidth			
	Security	SSH & Syslog: trigger on defined and unknown entries			
Software	Daemons	Up/Down functional tests Logfile entries			
	Applications	Availability Version Dependencies			
	Service	Up/Down functional tests Log file entries			
	Scientific Software	Up/Down			

Figure 2 Sample Monitor Parameter Table for the Server/Workstation and Network Node Classes.

3.3 Finding the SNMP OID

Both the monitoring systems for the ASTRI interface with devices use the SNMP protocol, so it is very important to know the mapping between the parameters (nodes) that we are interested to monitor and the SNMP-Object IDentifier (OID). The OID is basically the MIB address of the data item in the connected device that the SNMP manager requests from the agent. Each node monitoring parameter must to have is OID and a good way to know the OID is to use MIB discovery tools such as:

- Command line tool: ***snmpwalk*** linux, mac and windows
- GUI Based: ***MIBbrowser*** Java (linux, mac, win). ***PowerSNMP*** only Windows.

We used these tools to explore the MIB of all devices and to verify the SNMP Daemon (SNMPD) service configuration in the server, for example:

- We started reading tests of the system parameters (Figure 3).
- We tested the possibility to monitor the hardware parameters (Figure 4).

For mapping the physical parameters in the Servers MIB we simply install the application lm_sensors and make some simple configuration; lm_sensors is a standard application for Linux.

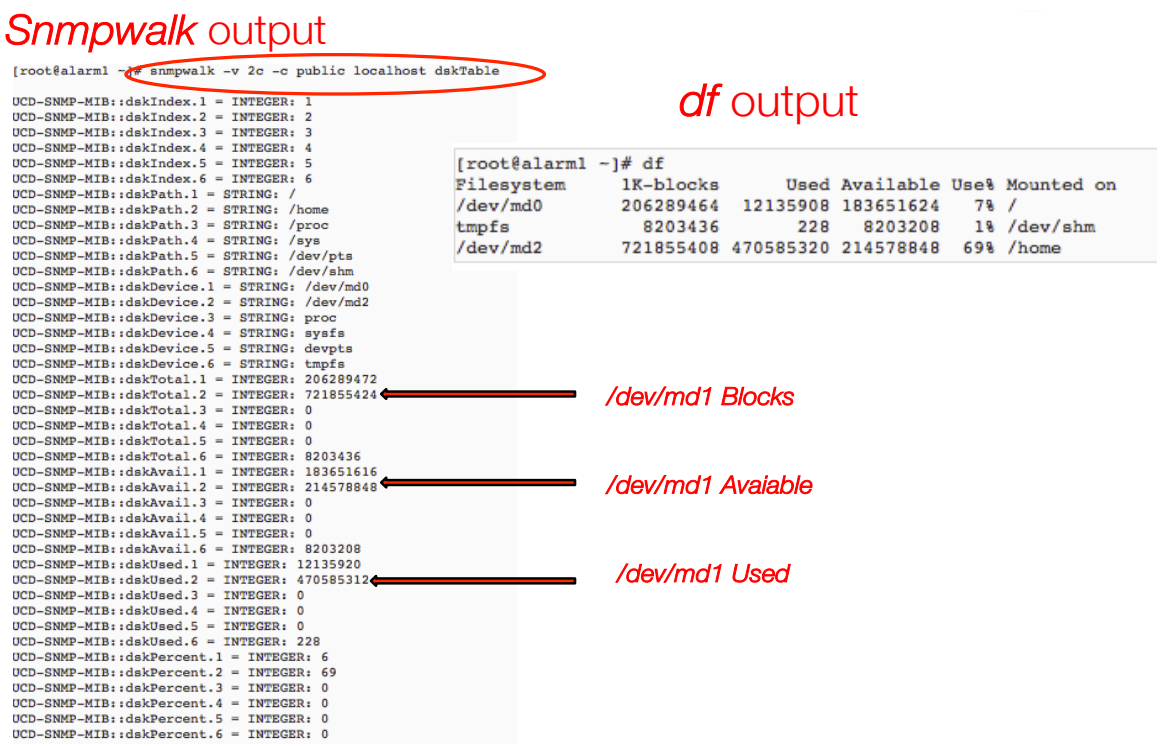

Figure 3 Using 'snmpwalk' to find the OID of system parameters: Software RAID status

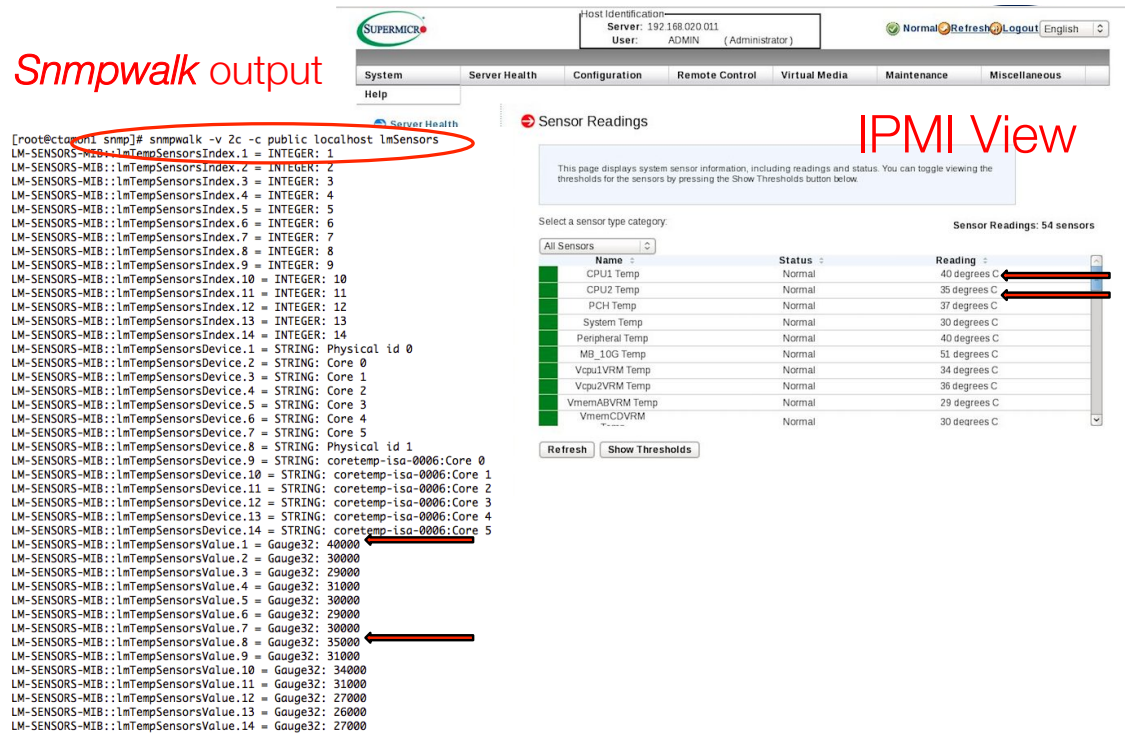

Figure 4 Using 'snmpwalk' to find the OID of hardware parameters: CPU's temperature.

4. TRADITIONAL MONITORING SYSTEM FOR THE COMPUTER CENTRE

This monitoring system must be a tool that allows monitoring the infrastructure, 24/7: regardless of the state of the telescope, the ICT must continue to work and we have to know its status.
Serious failures can often be predicted in advance through a proactive monitoring system, which significantly increases the system availability. This approach requires the SNMP support by all ICT hardware. The monitoring system will have to consider only the most important parameters that can automatically and surely detect a problem. It will allow quick and easy identification of the probable failure.
This system should be independent from everything else including the ICT itself, in order to allow us to understand the infrastructure status in any condition.
The Traditional Monitoring System Concepts and Requirements are listed below:

- It will be based on SNMP/ICMP Protocol.
- Simple interface (WEB based).
- Very easy to use.
- Fast implementation.
- Non invasive data collection.
- It monitors all required parameters.
- Customizable alarms with error thresholds (Mail &SMS).
- Generate reports.
- Possibility to export data for further processing.

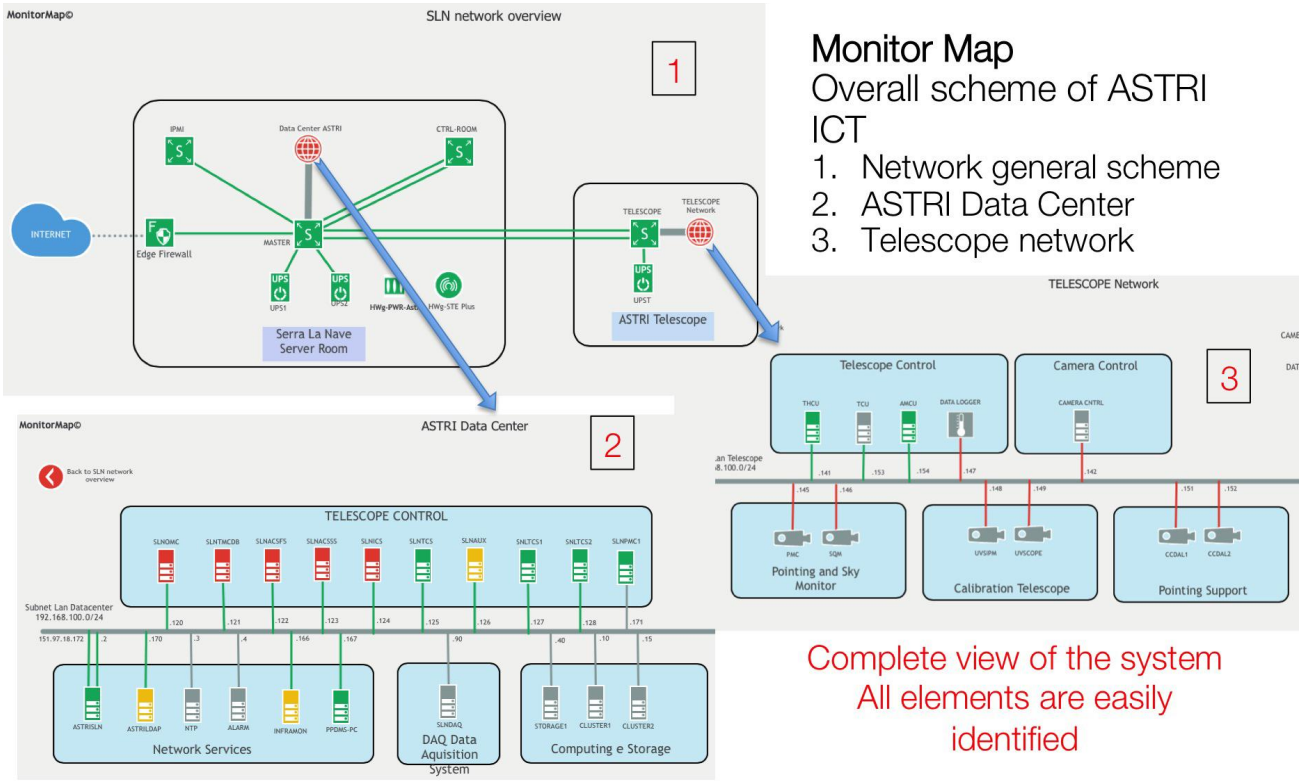

Figure 5 Main screens of the ICT ASTRI-SST-2M monitoring system

We started trying to implement the monitoring system with open source tools like: NAGIOS (https://www.nagios.org), OpenNMS (http://www.opennms.org), Ganglia (http://ganglia.info), but it is difficult to meet all the requirements.
In particular Ganglia is mainly dedicated to server cluster monitoring and it does not support well other devices, OpenNMS instead is primarily dedicated to network monitoring. NAGIOS is the most comprehensive system and it supports any type of device to be monitored. However the complete suite of NAGIOS management tools requires invasive plugins installation on the nodes and the monitoring communication is not only based on SNMP as we require.

In any case for all the three frameworks the integration into a single, simple interface it found to be very difficult if not impossible.
In addition to the intrinsic difficulties linked to open source systems, the need of a ready to use monitoring framework for the Assembly, Integration and Verification activities leads us to select a specialized company, VEM Sistemi (http://vem.com), to design and implement in a short time a complete and reliable monitoring system (Figure 5). However this proprietary software has two serious limitations: users have to pay an annual fee for its use and it is not very flexible because of the difficult implementation for certain types of monitoring.

5. MONITORING SYSTEM DEVELOPED FOR ACS

ICT is not a separate part of the general MASS Software[13] architecture, and then monitoring and error/alarm management should be integrated in the overall software environment. We made the choice to use OPC-UA server tasks for collecting all relevant monitoring data from the ICT components. This is the same approach used to monitor and control all the telescope subsystems[8].
This monitoring system is mainly targeted at night assistant / end user of the observatory (operator), its purpose is to report and / or prevent the fault during and / or immediately before the observation phase.
For ASTRI-SST-2M ICT we will develop and use a set of OPC-UA servers that will manage the subset of values to be monitored by ACS[9 10 11], convert their SNMP data, and present them as OPC-UA node (Figure 6). For this scope a set of corresponding ACS components will implement an OPC-UA client by means of an ACS DEV/IO.

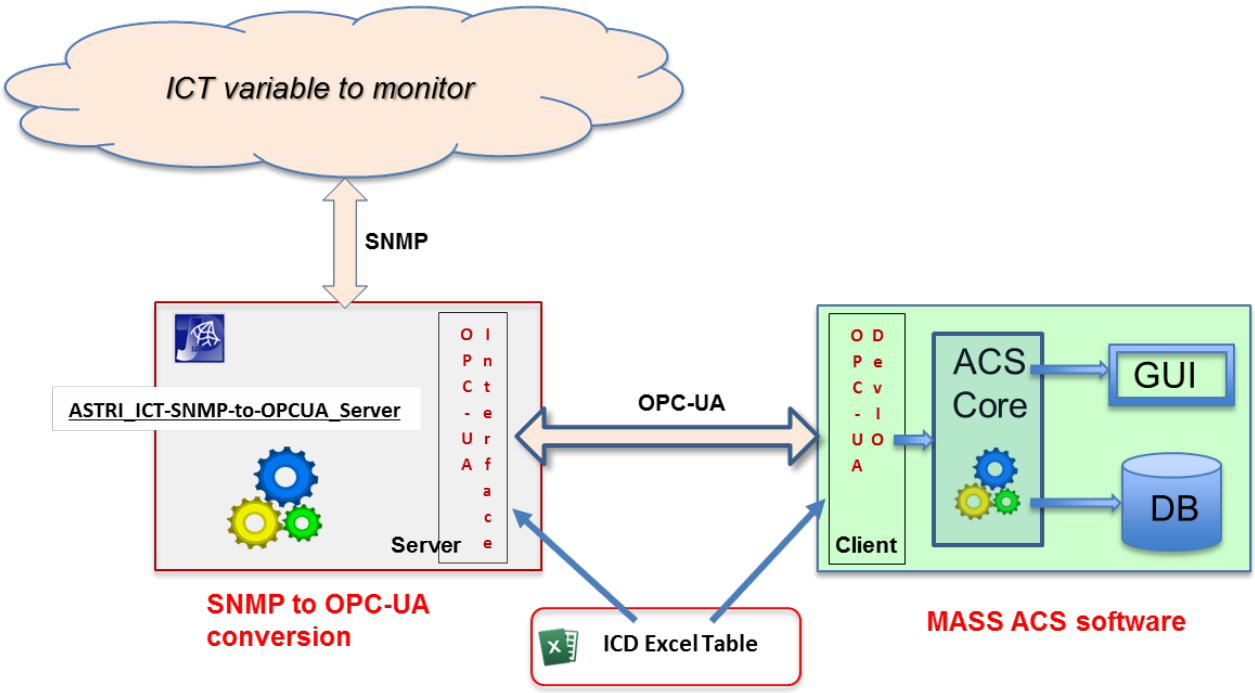

Figure 6 : SNMP to OPC-UA conversion

5.1 SNMP to OPC-UA Implementation work flow

We foresee the following steps (Figure 7):

1. Identifying the variables to be monitored for each element of ICT.
2. Identifying of SNMP OID for this variables, and then defining a mapping of these nodes in OPC-UA.
3. Writing an ICD for each ICT component to be monitored.
4. Using the ICD to generate the SNMP to OPC-UA server (with semiautomatic tool).
5. Generating basic OPC-UA client components for ACS (with automatic tools), starting again from the ICD.

6. Reading of the data supplied by the client OPC-UA in ACS and storing part of them in the ACS DB.
7. Displaying on the ACS GUI.

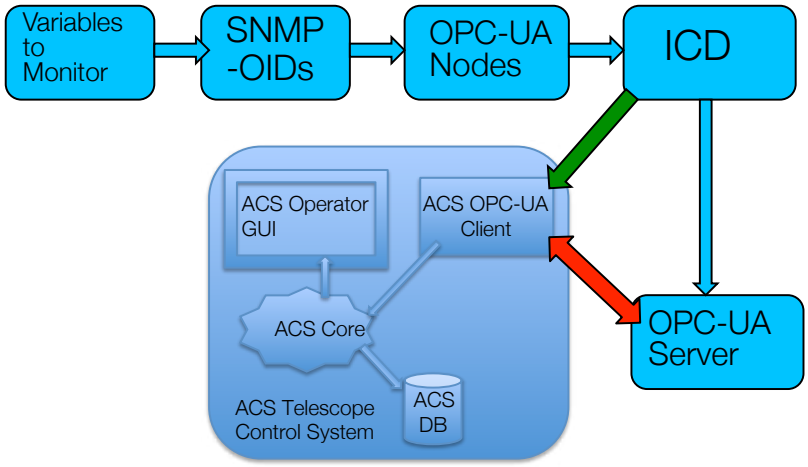

Figure 7 Implementation Workflow

The first two points of the workflow have already been discussed and are shared with the engineering monitoring. In the following we will analyze the other steps.

5.2 ICD for ICT component

For the ASTRI project (as well as proposed for the CTA array control general architecture), the Open Platform Communications - Unified Architecture (OPC-UA) technology controls and monitors all the hardware components. For this reason we describe all the OPC-UA Nodes and Commands with interface control documents (ICD) Excel tables, as shown in Figure 8.

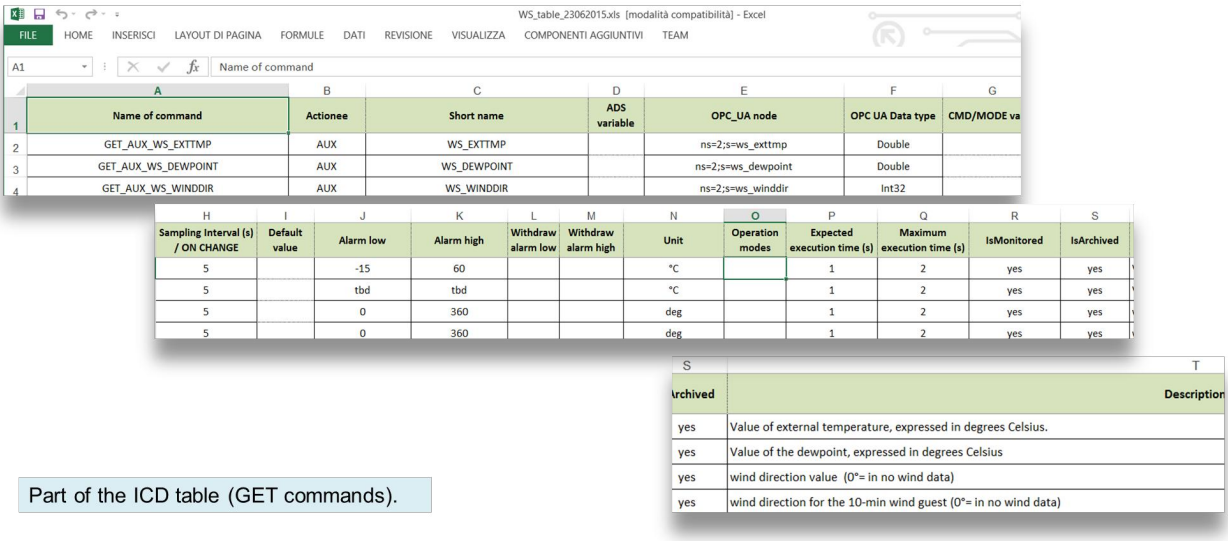

Figure 8 Interface Control Document (ICD) Sample

For the monitoring of ICT we use an extension of the ICD tables, adding some column such as IP address, SNMP-OID code, localization, GroupName, etc.

5.3 The SNMP to OPC-UA server

The basic idea is to develop an OPC-UA server specialized for each protocol (ICMP, SNMP) to be configured through an ICD table . The OPC-UA server is a runnable jar file that accepts as command line the ICD table file name.
Inside the server program we have inserted an excel parser in order to read the ICD table and to configure the server interface.
For each SNMP IP address we have developed an SNMP agent that is able to read data related to an OID parameter and convert this into an OPC-UA node on the other side.

The servers from SNMP/ICMP to OPC-UA protocol was realized using the tools listed below:

- OS: windows (xp, 7, 8.1, 10), linux.
- Language: Java SE Development Kit 8.
- Server opc-ua library: Prosys-OPC-UA-Java-SDK-Client-Server-Binary-2.1.2-478.jar, Opc.Ua.Stack-1.02.336.12.jar.
- Client opc-ua: ProSys Java Client 2014, UAExper opc-ua client.
- Eclipse IDE for Java developers – Version: Mars.2 Release.
- ICMP library: icmp4j.jar Open Source, LGPL license.
- SNMP library : SNMP4j - Free open source API for java.
- Apache POI - the Java API for Microsoft Documents.

5.4 Generating basic OPC-UA Client components for ACS from the ICD

In the MASS ACS Software side, using the Automatic Code Generator[12] with ICD input, one or more ACS components could be automatically generated. So each OPC-UA variable can be mapped as an ACS Basic Control Interface (BACI) property and no additional work is necessary to exploit the monitoring capabilities of ACS. ICT related data could thus be collected and analyzed along with all the other monitoring data from the various array subsystems Figure 9.

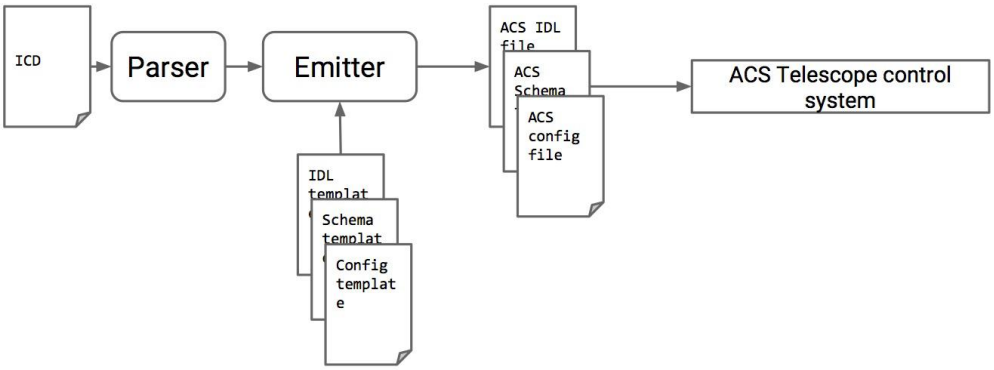

Figure 9: From the ICD to the ACS Code

6. ICMP TO OPC-UA SERVER IMPLEMENTATION

This is the first and most important OPC-UA server for the ICT monitoring because it determines whether a device or computer is UP or DOWN on the network. The server maps single device ping information (Ping is a computer network administration software utility used to test the reachability of a host in an IP network) in an OPC-UA Node. The OPC-UA Server is a Java program containing:

- An ICD parser for the appropriate reading of an ICD file.
- One ICMP agent for each IP-device to ping.

6.1 ICMP to OPC-UA Implementation work flow

The ICMP to OPC-UA server implementation workflow is a little different from the SNMP to OPC-UA server, we foresee the following steps, with the main differences between the two implementations highlighted in bold:

1. Identifying the variables to be monitored for each element of ICT => **UP/DOWN equipment State.**

2. **Identifying IP Addresses**, and then defining a mapping of these nodes in OPC-UA.
3. Writing an ICD for each ICT component to be monitored.
4. Using the ICD to generate **the ICMP to OPC-UA server**.
5. Generating basic OPC-UA Client components for ACS (with automatic tools), starting again from the ICD.
6. Reading of the data supplied by the Client OPC-UA in ACS and storing part of them in the ACS DB.
7. Displaying on the ACS GUI.

The server was tested on a dedicated Linux Virtual Machine (VM). Figure 10 shows part of the ICD table used to configure the OPC-UA server and the screen shot of the Prosys OPC-UA client that we use to test the server.
You may see some results from ping on IP nodes (slnics server: IP -192.168.100.124) and some variable-nodes implemented on the OPC-UA server.

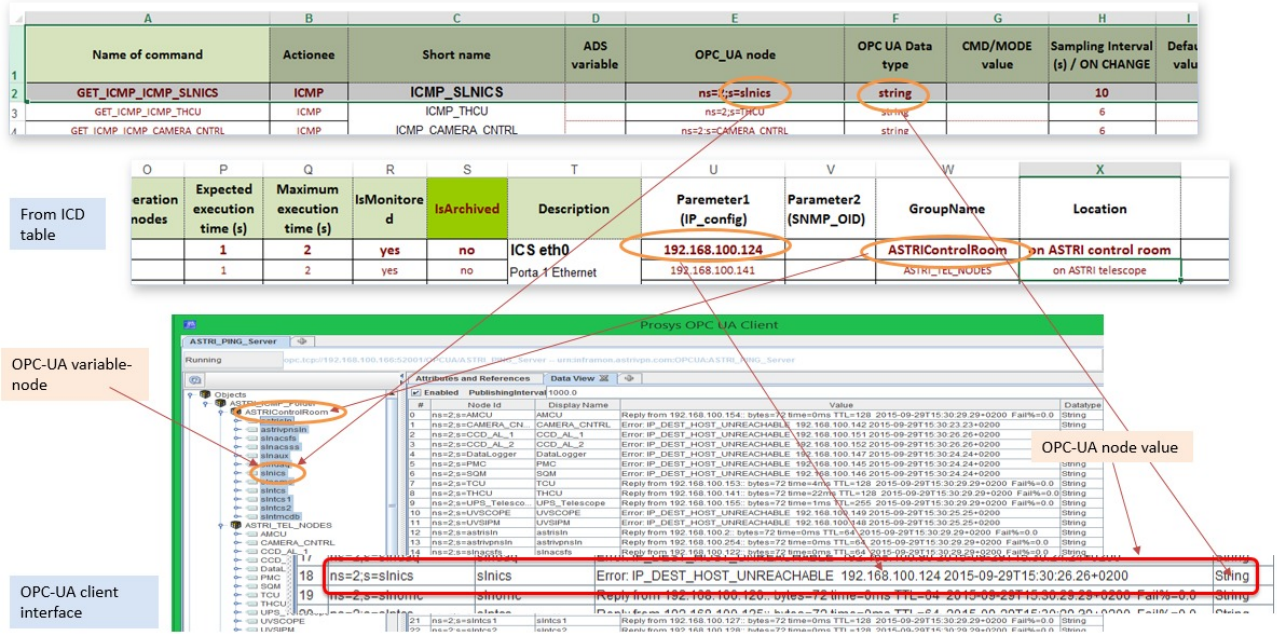

Figure 10: Part of the ICD table and a screen shot of the OPC-UA client

7. CONCLUSION

Despite the ICT monitoring system of the ASTRI SST-2M prototype being in an advanced stage and already running in some parts there is still much work to do to make it definitive and expand its concepts to the ASTRI mini-Array. The first important action is the complete definition of all the parameters to be monitored and the error levels to be assigned, as well as definition of the intervention procedures. This will allow having an accurate SNMP interface on which to place the two monitoring systems.
For the traditional monitoring system, we will have to find a solution that would allow us to have our own, possibly open-source and flexible, software to be developed autonomously by the ASTRI Project ICT group or with the help of industry. We have already begun to study and test the monitoring framework SANET of Marconi Labs (https://sanet.labs.it).
For the parallel ACS-based monitoring system the implementative lines and main modules have already been precisely defined. We have now to engineer the project. In particular we plan, in the near future, to identify a subset of significant parameters to be monitored, write the ICD, and prepare with these ICD an adequate number of SNMP-to-OPC-UA servers, and finally realize a GUI.

8. ACKNOWLEDGMENTS

This work is supported by the Italian Ministry of Education, University, and Research (MIUR) with funds specifically

assigned to the Italian National Institute of Astrophysics (INAF) for the Cherenkov Telescope Array (CTA), and by the Italian Ministry of Economic Development (MISE) within the "Astronomia Industriale" program. We acknowledge support from the Brazilian Funding Agency FAPESP (Grant 2013/10559-5) and from the South African Department of Science and Technology through Funding Agreement 0227/2014 for the South African Gamma-Ray Astronomy Programme. We gratefully acknowledge support from the agencies and organizations listed under Funding Agencies at this website: http://www.cta-observatory.org/.

9. REFERENCES

[1] Acharya, B.S., et al., "Introducing the CTA concept", Astroparticle Physics 43, 3–18 (2013).
Actis, M. et al., "Design concepts for the Cherenkov Telescope Array CTA: an advanced facility for ground-based high-energy gamma-ray astronomy", Experimental Astronomy 32(3) 193-316 (2011).
[2] Pareschi, G., et al., "The dual-mirror Small Size Telescope for the Cherenkov Telescope Array", Procs. 33rd ICRC, ArXiv:1307.4962 ,(2013).
[3] Pareschi, G., et al., "The ASTRI/CTA mini-array of Small Size Telescopes Dual-Mirror: a first seed for the Cherenkov Telescope Array", Proc. SPIE Astronomical Telescopes + Instrumentation 2014, Proceedings of SPIE 9145, (2014)
[4] Tosti, G., et al., "The ASTRI/CTA mini-array software system", Proc. SPIE Astronomical Telescopes + Instrumentation 2014, Proc. SPIE 9152, doi: 10.1117/12.2055067, (2014).
[5] Leto, G., et al., "The Site of the ASTRI SST-2M Telescope Prototype: Atmospheric Monitoring and Auxiliary Instrumentation" (arXiv:1402.3515), in proc. of "AtmoHEAD 2013 workshop / Atmospheric Monitoring for High-Energy Astroparticle Detectors".
[6] Douglas, M. et al "Essential SNMP", 2nd Edition, O'Reilly Media (2005).
[7] Mahnke, W. & Leitner, S.H., "OPC Unified Architecture - The future standard for communication and information modeling in automation", 3/2009, ABB Review, 56-61 (2009)
[8] Oya, I. et al. "OPC UA Guidelines", CTA document, ACTL-PM/140401
[9] Chiozzi, G., et al., "Corba-based common software for the alma project", in Proc. SPIE 4848, 43-54 (2002)
[10] Schwarz, J., Farris, A., and Sommer, H., "The alma software architecture", in Proc. SPIE 5496, 190–204 (2004)
[11] Jeram, B., et al., "Generic abstraction of hardware control based on the alma common software," in ADASS XIII ASP Conference Series, Vol. XXX (2004)
[12] Tanci, C., et all "Software design and code generation for the engineering graphical user interface of the ASTRI SST-2M prototype for the Cherenkov Telescope Array," Paper 9913-138, these proceeding SPIE, (2016).
[13] Tanci, C., et all., "The ASTRI mini-array software system (MASS) implementation: a proposal for the Cherenkov Telescope Array," Paper 9913-93 these proceeding, SPIE (2016).
[14] Conforti, V., et all., "Software Use Cases to Elicit the Software Requirements Analysis within the ASTRI Project," Paper 9913-148 these proceeding SPIE, (2016)
[15] Gianotti, F., et all., "Information and Communications Technology (ICT) Infrastructure for the ASTRI SST-2M telescope prototype for the Cherenkov Telescope Array," Paper 9913-137 these proceeding SPIE, (2016)
[16] Bulgarelli, A. et al., "The On-Site Analysis of the Cherenkov Telescope Array," ICRC 2015 Proc., Sept.(2015).